VOL. 1, NO. 1 SEPTEMBER 1985 $5.00

S/T

SIXTEEN/THIRTY-TWO APPLICATIONS

In This Issue: Page:

Editorial 03
Reader Comments 04
Byline Atari 07
First Subscriber Award 11

Departments:

You Can Do It!:68K ...by Dave Bessey 05
Dragon FORTH ...by Albert Lew 09
MIDI Beat ...by Richard Kaller 12
Walk Thru The Forest .....
An Introduction to Telecommunications
by Bill Petry & Bill Marlin 14

Reviews Etc.:

Rumors 08
Available Software 18
ST User Groups 18
Book Reviews 19
Coming Attractions 19
Software Review ...
Express Letter Processor 20

Meeting the Special Needs of Atari Programmers

>>>
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

S/T SIXTEEN/THIRTY-TWO Applications

William Petry, Managing Editor

David Bessey, M68000 Editor
Richard Kaller, MIDI Editor
Mikkel Lawrence, Communications Editor
Albert Lew, FORTH Editor
Chris Robertson, Pascal Editor

Bill Marlin, Technical Advisor

>>>
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

S/T SIXTEEN/THIRTY-TWO Applications is published monthly by William Petry. Copyright (c) 1985 by William Petry. No material may be copied without prior written permission of the copyright holder. Copyright is retained by the author where noted.

Subscription rate is $25.00 for 6 issues (paper).
$50.00 for 6 issues (disk).
Add $15.00 for subscriptions outside the United States.
Foreign subscriptions must be paid in U.S. Dollars. All issues sent First Class Postage.

Single issue prices:
Paper editions $5.00
3.5" diskette editions $10.00

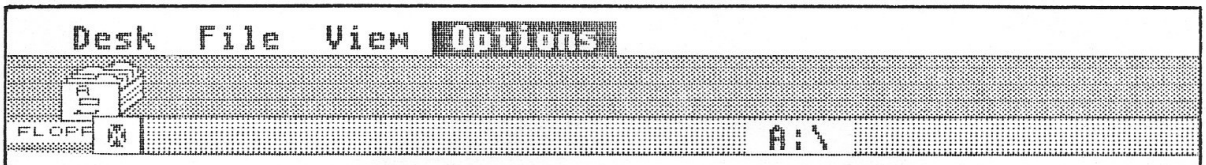
Quantity discounts: (of same issue)
15% for 5 to 9 copies
25% for 10 or more copies

Terms: Prepaid: No returns. Defective or damaged diskettes will be replaced upon receipt.

Please address all editorial and advertising inquires to:

S/T
SIXTEEN/THIRTY-TWO Applications
10760 Hwy 116
Forestville, California 95436

>>>
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<



Editorial!

With this issue of S/T Applications, we will be the first periodical on a diskette. Two factors decided this fate: Time and money.

Our ST computers came into our lives with more promises than a presidential campaign; more rumors than a movie star's love life and out of it all, we have our idle computers and ourselves.

The idea behind S/T Applications occurred to me one night in May after having read some 'fluffy' articles on the ST. Such a powerful computer that few people will really be ready to cope with it. I mulled over the information (or really the lack of it) that I'd just looked at and I knew then that the ST could never be adequately represented by the existing Atari-oriented magazines. It's a whole 'nother ball game. The microprocessor, capable of addressing 16 to 32 Megabytes of memory; screen resolution, capable of displaying satellite weather pictures clearly; GEM graphics environment management system that allows only previously imagined ease of roaming around the screen. The list goes on and on.

The following day I got together an outline of where, when and how to get this idea into reality. With the promised June delivery date. I was pressed for time to get the ball rolling by the 15th of July release date I had in mind. It didn't take long to realize the magnitude of the problem I had at hand. Without a computer I had nothing to write about. Without software I had nothing to write it with (other than my Atari 800). So the slow grinding wheel of reality wore away on S/T Applications.

Finally in July the computer, TOS, and LOGO arrived. After formatting 20 or so diskettes, my ST became a conversation piece..."Look at my new computer...It doesn't do anything...but it does look really nice." I had never felt so helpless in my entire life about a computer. Putting out the $799.95 didn't make me feel bad, it was the unuseableness of the whole thing that hit me. This wasn't alleviated in early August when we bought the ST development package. We were excited driving down to Atari in Sunnyvale to pick it up after a week or so of delays from the printers.

However, scanning over the documentation showed that there wasn't an editor in the package. Another pang of reality. With my Atari 800, every software package came with its "built in" editor. It never occured to us that we'd need an editor. A week later, after meeting Wendel Brown of Hippopotamus software at the August A.B.A.C.U.S. meeting at Fort Mason, San Francisco, CA, we were on the road to getting an editor with a review copy of the Hippo 'C' Compiler. Our hopes were up, but Haba's problems with label printers and subsequent relocating to a new building, even this helping hand has yet to reach us.

Finally, on the last day of August, we got a copy of the MicroEMACS editor from Atari (part of our development package), and so could finally begin composing our first articles on the ST.

On Tuesday, September 4th, Mikkel Lawrence and I went down to the San Leandro Computer Club's meeting. There we met Bob Barton, the club's president. I joined the club, and in the process, made some new friends thru whom we got STerminal. This public domain terminal program got us our first ability to transfer text files to the ST. Delays and problems with the Atari 800 KERMIT have yet to be overcome, so Sterminal was a great step forward.

Albert Lew had already had 4x Forth for some time and Richard Kaller had over a year of working with Synthesizers (on the Musician side) and several months earlier had gotten a MIDI interface for his Atari 800. Dave Bessey has been a top notch 68000 programmer for several years now, and he needed only to get his ideas over to the ST. These good qualities were right for what I needed to get my idea into reality.

Getting this first issue out to you by September 16th is what I mean by 'timely distribution of information'. It means that when your disk copy of S/T Applications is mailed, the information is up to date as of the previous day. Not 1 to 2 months old.

Most periodicals bend with the pressure of advertisers. This is the 'Money' end of the deal. Rather than have flowery articles dealing with only

half-useable software and hardware items, I want to make sure that the product 'speaks for itself'. The past few years have shown me that Atari computerists have been at the mercy of the commercial software producers. I guess that Pong and the 2600 were both Atari's boon and bane. The 800 bears the stigma of "game machine" which, no one familiar with it would agree. The misconception is still there (did you hear it when you showed off your ST?).

We have a great computer. We must take the initiative and look out for our own best interests. Creative innovation is what its all about.

Bear with us these first few issues. The disk-based periodical has just been born and I'm sure we'll have a few diapers to change before it finally grows up.

Bill Petro
Managing Editor

Reader Comments

For those of you who didn't see Computer Current's Aug 27-Sep 9 issue: On page 43 they tell how some insider at Atari sneaked these four icons in among the control characters as a joke. They make a picture of J.R. "BOB" Dobbs of the Church of the SubGenius. He is the invention of some art students, who wanted to create a bogus religion based on flying saucers, lunatic conspiracy theories, and the occult. One of his sayings was "F - 'em if they can't take a joke."

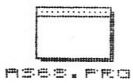
If you check pages 63 and 65 of your sourcebook for Atari (ST) LOGO, you'll see these icons as chars 28 through 31, if you look carefully. They're just to the right and above the two that make the Atari symbol. Study the listing to see the difference between print and type.

```
TO BOB
REPEAT 35 [TYPE CHAR 28 TYPE CHAR 29] PR "
REPEAT 35 [TYPE CHAR 30 TYPE CHAR 31] PR "
PR [F - 'EM IF THEY CAN'T TAKE A JOKE.]
END
```

You can also print out the Atari logo with this short routine:

```
TO ATARI
REPEAT 25 [TYPE CHAR 14 TYPE CHAR 15 TYPE CHAR 32]
END
```

Lois Hansen
Oakland, CA



by Dave Bessey

You Can Do It!

(c) 1985 by David Bessey

"You Can Do It!" will present information on assembler and machine-language programming of the 68000 series MPUs. One topic will discussed each month; let us know which topics are important to you. The information in these articles will be generally computer-independant: it will run on ST, Amiga, HP Series200, and others. Anything which IS computer-dependant will be labelled as such. Standardized abbreviations will be used throughout, with non-standard terms clearly labelled. A word-list is included with this article. I recommend that everyone seriously interested in M68000 programming get the book "M68000 16/32-Bit Microprocessor Programmer's Reference Manual"4thEd from Prentice-Hall, Inc.,NJ. ISBN 0-13-566795-X, 1984.

Later on there will be some discussion of hardware. Again, it will be computer-independant or else it will be clearly marked. All digital signals will use the names given by the manufacturer of the device being discussed with the exception that the signal-names will be prefaced with a lower-case 't' or 'n' to indicate whether the signal is 'true' or 'false'; examples will be shown later.

This month's discussion is about hardware address space and instruction(software) addressing modes.

The MC68000 has 16MB of address space (the MC68008 has 1MB, the MC68020 has 4096MB) because it has output lines for twenty-four bits of address; these are the pins labelled tA23 through tA00. Three additional lines, tFC2, tFC1, and tFC0, if used, allow eight address spaces to be used. (That's 128MB!) Only five of these spaces are actually used. The FC outputs are assigned as shown below.

0	↕	↕ Reserved by Motorola
1	User data	
2	User programs	
3	↕	
4	↕	
5	Supervisor data	
6	Supervisor programs	
7	Interrupt acknowledge	

"Supervisor state" is used for Operating System programs. "User state" is for application programs. The segregation of data and program spaces allows for re-entrant programming. The Interrupt Acknowledge space is for vectored interrupt processing. These subjects will appear in future articles.

The M68000 family have seven (standard) addressing modes and up to eight (five in use at this time) 'special' addressing modes.

The mode & register field is six bits: three for the mode, three for the register.

Mode	address operation
0	Data register contents
1	Address register contents
2	Memory: An points to data
3	like M2; increment An after use
4	like M2; decrement An before use
5	like M2; add a sign-extended 16b value.
6	Indexed (see index discussion below)
7	--special-- reg field is new mode number
0	16b absolute (000000:00FFFF)
1	32b absolute ((00)000000 to (FF)FFFFFF)
2	PC+16b sign-extended displacement
3	PC+index (see index discussion below)
4	Immediate data
5	\
6	->reserved for future use
7	/

Indexed instructions of the M68000 two extra bytes to provide additional address information. These two bytes are formatted as shown below:

bits	content/meaning
15	index reg type: 0=Dn, 1=An
14:12	register number (A7=stack)
11	index size: 0=16b sext, 1=32b
10:08	000
07:00	8bit sign-extended integer

Abbreviations/Glossary

M68000

abs absolute

absolute An absolute address is one which never changes: it is not indexed, does not have a register or displacement added to it. see--relative

ADDR address

An Address register. The 'n' is a number 0:7. A7 is Stack Pointer.

b bit or bits

B byte or bytes

displacement A value contained in an instruction which is added in during address calculation. It is used to provide an offset into a table or list. The displacement is a constant integer specified when the program is written.

Dn Data register. The 'n' is a number 0:7.

HP Hewlett-Packard Company. Their Series200 uses M68000 parts.

Immediate Immediate data are included in an instruction, and are used to load constants into registers.

FC Function Code outputs of the M68000 family.

inst instruction

M From 'mega' meaning "large"; one million(decimal), 1048576 binary (2^20).

m Milli; 0.001 (1/1000)

page 4096 bytes (16 sectors)

PC Program Counter; this register keeps track of where the computer is in the program it is running.

reg register

relative Relative means that one address is specified by referring to a different one. A statement such as "Goto LOCATION+34" is relative because "LOCATION" can change its value when the program is reloaded. "LOCATION+34" is always 34B after "LOCATION" regardless of where the program is in memory.

sector 256 bytes

segment 16 pages, 256 sectors

sext sign-extended

sign-extended The M68000 uses 32b values in its address calculations. If you provide it with an 8 or 16 bit value it will copy the left-most bit into the remaining 24 or 16 bits (on the left) to create a 32b value. This process is called sign extension because the leftmost bit of a 2's-complement number is its sign (0=positive 1=negative).

ST Atari Corporation's ST series.

Two's complement 2's-complement numbers are designed so that when you add a negative and positive number the effect is the same as subtraction. This process simplifies computer hardware because subtraction is more difficult than addition. To convert a number to its 2's-complement you need to invert all the bits in the number (this is the one's complement) and add 1. Therefore, 0000:7FFF is 0:32767 dec and 8000:FFFF is -32768 to -1 dec. There is no -0 or +32768. To subtract 3 from 5:

```
5: 0005 invert add1   0005
3:-0003  FFFC  FFFD  +FFFD
   ----              ----
2: 0002            (c)0002
```

(c) indicates a carry bit.

The two's complement process causes a negative number to have its leftmost bit be 1. see-- sign extension

Byline Atari

Neil Harris Live!

This discussion took place on 04 Sep 1985 between Neil Harris of Atari,Inc., and Bill Petry.

NH: Can I ask you a silly question? What is your background in the magazine business?

BP: You asked me that last time. I just read them. I've got a lot of feeling about magazines.... I majored in education and biology..with that point of view, I look at a lot of magazines and feel they're not telling me alot. So my impression is get the information out to people and they can come back and say it's there.... and have it really be there. Then you can take it step by step and really get something out of it.

BP: I did hear a rumor last night that you'll have a 1 meg, 260ST for Christmas?

NH: How can it be a 260 if it has 1 meg memory? What form that 260 will finally take nobody knows exactly yet, its probably almost as easy to make it 1 Meg as it is 260. Just different chips. I have really not heard inside the company any firm plans on when that will hit or what's going to happen with it.

BP: Is there any way I can get BASIC?

NH: Not yet. You don't want BASIC just yet, it's real close now. The problem with that you can't count on it (the current 'bugged' version) to much. I'll tell you what happened to me. I took the latest version of BASIC, as of the morning of the day that I did this user group meeting. I had a disk full of programs that had worked on the week's before version, and they wouldn't work any more. The reason was that they changed the parameters of the color command, and it was setting the wrong color register, and when I was pokeing around with the colors it didn't change anything. They are changing the way BASIC talks to the operating system, and that can lead to all kinds of subtle little things changing that you have to just work around. Most of the commands work, for the internals and disk stuff. But a lot of the stuff with the GEM interface will change.

BP: How much memory will BASIC address.

NH: It can access anything. The only reason that the 8-bit BASICs were originally designed for 64k is that that's all the chips could access back when those BASICs were being written, so it was the upper limit. There's no inherent limit in the BASIC itself, they've designed it so that whatever's there is accessible. What I'm really hoping, but there's no firm plans for this yet, is that if they can get BASIC down to 128K, they could ROM that and make it a plug-in cartridge. Then you'd have a tremendous amount of free memory. But there's still debate as to whether they really want to do that because of the cost of putting it into ROM, there's really nothing extra we'd get out of it.

BP: What are those 'eyes' with the rays 'looking' down that appear on the screen when you hit a Control 'C'?

NH: You mean mushroom clouds. They mean something blew up. The number of mushroom clouds determine which error you got.

BP: Is that listed somewhere in the manual?

NH: No. It's an internal system error, it's nothing that would do you any good. It means that when whoever wrote the program that did that to you, they didn't forsee the possibility of a control 'C', and so they didn't make sure you had a clean exit, so the O.S. yanked you out of the program and gave you some kind of error code.

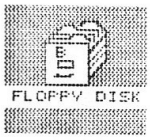
BP: How many can you expect as a maximum?

NH: It depends. I got caught up in a loop once and got a whole line of them. Theoretically you can get up to 16, as legitimate messages. But if something went more haywire than our guys would have expected, you can get an awful lot of them. That would be an interesting contest. How can you get the most mushroom clouds.

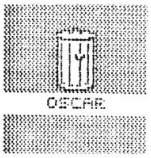
NH: You haven't worked much with LOGO, I take it.

BP: I've never tried LOGO before. My wife tried a bit last week and quickly gave up.

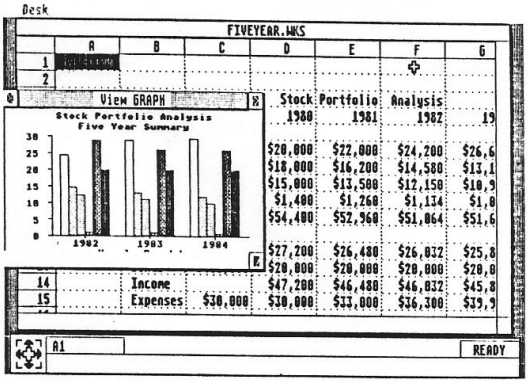
NH: Go to a book store and pick up a book from SAM's, it's from the Waite group, on Apple LOGO. It's really good, and it doesn't just do Turtle graphics. It deals with list processing, and some of the more powerful stuff LOGO can handle. If you could do some good programs on how to do the real...I mean you can do artificial intelligence programming just like you can with LISP, except that I've never seen anybody expain how you do it. I know its there. I don't know how to do it myself. I know its in the language. Leonard Tramiel could do it if he had nothing to do for a couple of weeks. So that might be an area where you could help a lot of people who are sitting out there with LOGO, trying to make it do something.



Rumors



VIP Professional, VIP Technologies (of Goleta), 132 Aero Camino, Santa Barbara, CA 93117 (805) 968-9567: Dave Nelson said that this Lotus 1-2-3 clone would be released at the end of August. As of the first week of September, Dave said that due to the upcomming revision ('E') of TOS, this package would not be available until later on this month. They want to make sure the revision doesn't play havoc, even though they stuck to the guidelines given to them by Atari. (04 Sep 1985)



VIP Professional

Valdocs, Rising Star Industries, 25500 Hawthorne, Suite 2000, Torrance, CA 90505, (213) 373-9112: This is an assemblage of 14 VALuable DOCument units, including wordprocessor, data base, spreadsheet etc. See the September 1985 issue of The Computer Shopper for a review of the EPSON computer version of VALDOCS. Valpaint and Valdraw are to be released the first part of October.(10 Sep 1985)

HABA products, HABA systems, 15154 Stagg Street, Van Nuys, California 91405-1025, (1-800-367-4222) to order. Their 'Solutions' series, which includes Business Letters & Wills, will be available this week, at a price of $49.95 ea. HabaWord, HabaFiles, HabaCalc & Graph, HabaCheck Minder, HabaComm, and the Hippopotamus 'C' Compiler are in the offing at a price of $74.95 ea. (14 Sep 1985)

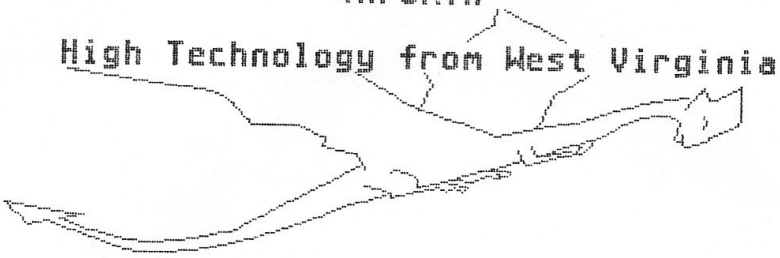
GEM Write (take a look at it on the IBM PC at your local IBM shop), and BASIC should be out in about 2 weeks (have you heard that before?).

by Albert Lew

The DRAGON Group, Inc. presents

4xFORTH

High Technology from West Virginia



A FORTH Overview

Have you finished investigating the shallow waters of GEM? Is your DR Logo disk starting to collect dust? If that wonderful piece of machinery, your incredibly powerful, incredibly stagnant ST is just longing to be programmed, wait no longer. Dragon Group's FORTH is available now for $99. But why purchase FORTH, instead of Basic, which you are already familiar with, or C, the language many ST enthusiasts are raving about?

Basic's faults are well known, and FORTH has almost none of them. FORTH takes up less memory than assembly language, executes faster than everything short of machine language, is much more enjoyable than Basic, and isn't beset by other Basic problems like practically incompatible dialogues and the infamous 8 bit Atari Basic 'lock-up'. What about the much vaunted C language? For starters, C takes up much more memory than machine language and can gobble up almost as much RAM as Basic. C executes decently fast, and isn't beset by nagging problems, but there is one big hangup with C. The word is compile. In essence, compilation is the process of taking your C program and translating the C code into machine language. Would you like to create a simple program in C to greet you with the text "I am an Atari ST computer."? Prepare coffee and comeback in six minutes with Digital Research's C. Or disk load LodeRunner on your 800 while Hippo C compiles. With FORTH, you have a compiler and an interpreter. To create this program in Basic would require maybe fifteen seconds to enter. Type run, and you're off. With FORTH, the same fifteen seconds would be needed. For a C program, this quarter minute would be added to the compilation time.

FORTH's concept of compilation and interpretation what makes FORTH so unique. With FORTH, programs are not written to be compiled or interpreted linearly, but in blocks. When you write a Basic program, you tell the computer what to do with details which are Basic words such as POKE, or PRINT. These details strung together can be thought of as a line which represents your program. You may call several routines with GOSUB, but you are really just moving around parts of the line. The same philosophy applies to C, Pascal, Fortran, Cobol, Modula, and even assemblers. FORTH, however consists of FORTH words which you use to create reusable building blocks. When you put the building blocks you have created together, you create even more building blocks ad infinitum. You can create a 'construction set', a large block that is your program, or maybe something in between.

FORTH words include 'FORTH-style' math functions, Boolean (comparative) operations, loop control, I/O handling, greatly enhanced relatives of PEEK and POKE, variable handling, and an assembler and assembler support. These may sound fairly standard aside from the assembler, but how these words are used in the FORTH philosophy gives them their hard hitting punch.

If you are a software developer, FORTH can speed development time by up to a tenfold factor, and if you just tinker and experiment with the guts of your ST, FORTH will allow you to do so painlessly. For instance, if you were developing software in C and found a misspelling in a subroutine, it would be convienient to recompile the subroutine and go back to work. In C, however,

things don't quite work that way. Instead, you start your ST compiling, go out to lunch, and hope 250K of source code compiled by the time you get back. With FORTH, recompile the FORTH word with the error, and the misspelling is fixed. There is no need to recompile other words that contain the word with the misspelling. If your misspelling is in the main routine (C), or word (FORTH), it is apparent that a recompilation of the whole program is necessary. If you have C, you have to go out to lunch (again!). With FORTH, 300K will have finished compiling in the middle of a five minute coffee break! If you're not a commercial programmer, FORTH still holds a treasure chest for you. If you would like to PEEK and POKE around, you may do it using FORTH or the FORTH assembler. Most importantly, your results are immediate. With C, you would have to wait a minimum of forty seconds, just for a simple PEEK or POKE. Of course, Basic will get instant results, but a very temperate patience is required while executing any Basic program. If you would like to write your own 'Peek 'n' Poke Construction Set', FORTH would handle your needs nicely. Sadly, few novice C programmers could manage as well.

Now for the icing on the cake. If you don't like FORTH's syntax, you can create your own! If FORTH's equivalent of Basic's PRINT seems ugly, go ahead, change it to PRINT! If you are annoyed that FORTH lacks a statement implemented on C or Pascal, it won't be difficult to write a FORTH equivalent. If FORTH, has whetted your appetite, get ready to go FORTH!

Going FORTH

Once you have loaded up Dragon FORTH, you should see their I.D. etc. and get the 'ok' prompt with flashing cursor. If you have a demo, exit the demo and you will get the 'ok' prompt. If your cursor doesn't flash, you forgot to install FORTH as a TOS application. It is helpful to install FORTH and then Save Desktop. This way, FORTH will automatically be installed as a TOS application. Wait no longer, it's time for FORTH!

EXAMPLE 1

Type:
.(I AM THE BEST PC IN THE WORLD.)[RETURN]
(Don't type [RETURN], just hit the return key on your ST when you see [RETURN])
You see:
ok .(I AM THE BEST PC IN THE WORLD.) I AM THE BEST PC IN THE WORLD.

What happened is apparent; .( ) in FORTH produces the same result as PRINT "" in Basic. The space after the opening parenthesis is required, but no space is needed before the second parenthesis. If thIS space was omitted, FORTH would try to execute '.(I', which is not a legal FORTH word. The following example will clear up why the space is needed.

EXAMPLE 2

Type:
2 SPACES .(I AM AN ATARI 520 ST.)[RETURN]
You see:
ok 2 SPACES .(I AM AN ATARI 520 ST.) I AM AN ATARI 520 ST

The command 2 SPACES was used to show the purpose of the space in FORTH syntax. Three (remember, .(needs a space after it) spaces could have been inserted after .(, but the purpose of the space in FORTH syntax would still have been hazy. There are spaces between the number two and the word SPACES, between SPACES and .(, and between .(and I AM AN... because spaces are needed to separate FORTH strings (I AM AN...), numbers (2), and words (.(and SPACES). For now, the space can be thought of as FORTH'S equivalent of Basic's colon. Incidentally, a space in not required before) because FORTH looks for the close parenthesis after it has encountered .(. The purpose of 2 SPACES is clear, and 15 SPACES, 10 SPACES, and even 49 SPACES all produce similar results. To really understand 2 SPACES, it is necessary to know about the stack. The stack can be used for many things, as the following example shows.

EXAMPLE 3

Type:
69 68 67 66 65 EMIT EMIT EMIT EMIT EMIT[RETURN]
You see:
ok 69 68 67 66 65 EMIT EMIT EMIT EMIT EMIT ABCDE

EXAMPLE 4

Type:
11 3 + .[RETURN]
You see:
ok 11 3 + . 14

EXAMPLE 5

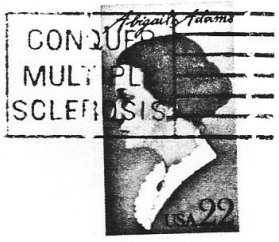
Type:
41 EMIT CR SPACE 40 EMIT[RETURN]
You see:
ok 41 EMIT CR SPACE 40 EMIT)
(

EXAMPLE 6

Type:
.[RETURN]
You see:
ok . . ? <stack!>

Whew! Well, that was a handful, wasn't it? Before I explain the example 3, an explanation of the FORTH stack would be helpful. For math oriented people, the FORTH stack uses RPN and LIFO (last in, first out), and works almost like an HP programmable calculator. For those not familiar with RPN, LIFO, or HP calculators, the FORTH stack is used to execute math and Boolean functions. The stack also serves as a number holding area for many FORTH words, including SPACES and EMIT. The stack can be thought of as a well where a column of numbers is placed. In the first example the numbers 69 through 65 appear in descending order. When FORTH reads the 69, it drops 69 into the stack. (Most FORTH programmers refer to this as pushing 69 onto the stack). When the FORTH interpreter reads 68, it drops 68 into the stack. Then FORTH reads 67, 66, and 65 and drops those numbers into the stack in that order. When you look down the well (stack), you see 65 on top of 66, which is on top of 67, and so on. 69 is at the 'bottom' of the well. The next word the interpreter sees is EMIT, which is the equivalent of Basic's PRINT CHR$(X);. When FORTH encounters EMIT, it pulls the top number off the stack, 65 in this example, and places ST ASCII character 65, or 'A', on the screen. The 65 is then discarded, and 66 is now on top of the stack. EMIT is now interpreted again, and the letter 'B' is placed on the screen and so on up to the letter 'E'. SPACES uses the stack to get numbers as well. SPACES pulls the top number off the stack an prints out that number of spaces.

Example 4 pushes (drops) the number eleven onto (into) the stack, and then proceeds to push the number three onto the stack. The FORTH word + simply takes the top two numbers of stack and adds them together. The two numbers are discarded and the sum is pushed onto the top of the stack. To see the result, the FORTH word . pulls the top stack number off, prints it to the screen, and then discards the number. Example 5 just introduces two new words, CR and SPACE. CR produces a carriage return (Dragon FORTH will not insert a carriage return when it gets to the end of a screen line) and SPACE is the equivalent of 1 SPACES. Finally, if you typed in the examples in the order that they appeared, you would find that there was nothing left on the stack by the time you got to example 6. But you tried to pull something (which was actually nothing) off the stack with . anyway. FORTH simply responded by giving you <stack!> to tell you that you had tried to do something with the stack that was physically impossible. The next column: CREATION OF THE WORD.



Yes it was a tie between Robert Floyd of St. Paul MN, and Paul Maternowski of Cedar Rapids, IA, for being the first subscriber to S/T Applications. Both letters were postmarked PM on 04 Sep 1985. Thank you both for being the rock of support we need to make our quest successful. Thanks again!

by Richard Kaller

MIDI Beat

By The Power Of MIDI

At last, the STs are finally here. Tear open the packing and gently lift my 520 out of the box. Now let's see what hardware made it to the production model. It has 512K RAM with a direct memory access rate of 1.3 meg/sec, an RS-232C serial interface, a Centronics interface for printers, a hard disk interface, a monitor port; and it still has MIDI, not as a peripheral, but with the hardware and software routines built in. For those of you who may not have heard of MIDI, it stands for Musical Instrument Digital Interface. MIDI is a standard, covering both hardware and software protocol, agreed upon and designed by musical instrument manufacturers, so that instruments such as synthesizers and drum machines could talk with each other. Because all of these MIDI equipped instruments are really computers, dedicated to music but still computers, MIDI also enables them to talk with personal computers like the ST.

What's this mean for you and me? Well, if you're a musician and you haven't heard about the power hidden in a computer, listen closely. Your ST computer can function as a multi-track recording setup for MIDI-equipped instruments, in a small space at home, simulating most of the capabilities of a commercial tape recorder, like autolocate, punch-in and punch-out, overdubbing, and track bouncing. The main difference is that MIDI is memorizing sequences of note-on and note-off commands, instead of audio signals quantitized to digital data. (More on the significance of this later.) To play what you have recorded, you simply reverse the flow of data with the computer feeding the information back to the voices in the synth or drum machine, rather than driving an audio signal through a conventional stereo system. With the ability to send parts addressed to 16 different channels over one cable, you can playback as many parts in real-time as you have seperately assignable voices.

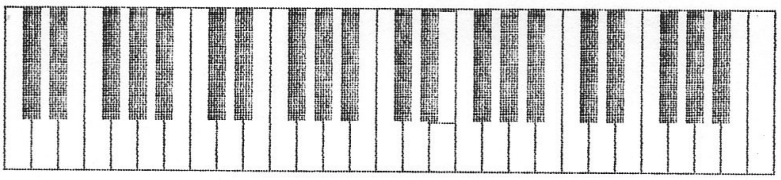
After recording, you can edit the music in much the same manner as a word processor lets you manipulate text. Repeat a phrase here, cut out a note there, even change the tempo or transpose the key; things that can't be done easily on tape. (Editing functions are also extremely useful in creating and maintaining patch libraries, blocks of data that define the tonal qualities of notes, as opposed to their pitch.) The next step is hard copy. A graphics printer can give you instant sheet music, complete with all the latest edits you've made. No more time spent handcopying music notation. No more trying to remember just what notes you played during that improvised solo everybody loved. On top of all this, you get accurate, high speed, easy access, massive storage for all MIDI data on disk. I think we all are familiar with the trials and tribulations of using cassette recorders for program and data storage.

What's next? If you're not a musician but always wanted to be, MIDI releases the tutorial abilities of computers to benefit young and old alike. A computer never gets grumpy having to correct beginners' mistakes and is discerning enough to catch any mistruck notes or the slightest fluctuations in tempo. Even if you can't play a lick, most MIDI recording software allows you to enter notes step-by-step to be played back in real-time. This is slow, but also valuable for correcting timing errors. Or if you want to sit back and listen, there are floppy disks of pre-recorded music; boot up the disk and voila! your very own computer-driven Muzak. Now, we can be composer, conductor, performer, and audience rolled into one.

This is great for musicians and music lovers, but what about those who can't carry a tune in a bucket (or don't want to)? What else does MIDI have to offer? Due to the nature of MIDI, it can be used to direct much more than music. Remember I said that MIDI doesn't use converted audio signals. MIDI code uses some nibbles (half bytes)

to define channels, a few more to turn notes on and off, and small blocks, maybe 16 or 32 bytes long, to simultaneously change all parameters. If you substitute 'events' for 'notes' and 'control panel settings' for 'patches', it becomes clear that MIDI could turn on or off any properly equipped device, or make adjustments to its controls. In fact, it is already being used to control stage lighting during performances. Since you can turn lights on and off onstage, why not in your home or workplace, according to some schedule or in response to external events? If you can control lighting, why not systems such as heating, air-conditioning, watering, security, and appliances. This is truly a situation limited only by our imaginations.

Future columns will deal more closely with the details of MIDI; how it works, where it has been, and where it is going. Interspersed in all this will be reviews of MIDI software, books, and existing instruments; previews of new controllers and controllees; interviews with programmers, musicians and technicians; and any other views that might be appropriate. I will also devote a column here and there to answer letters, so write me with any questions or comments. Because of MIDI's parentage, the emphasis for now will be on music, but you will be the one to say where we will go with the power of MIDI.



A MIDI DEMONSTRATION

As extras on this month's disk version of ST are MIDIDEMO.PRG and *.SNG files. The demo was written in FORTH (it requires the 4TH.BLK and 4TH.IMG files on the same disk to run) and makes use of OS routines to access the MIDI ports. It is included here with the kind permission of the good folks from ATARI. So, if you already have a synthesizer, or can beg, borrow, or steal one, you can have your ST playing music in no time at all.

First you need to determine if the synth you are using is MIDI capable. Almost all of the new keyboard instruments are MIDI equipped (and conversions are available for many pre-MIDI boards), but if you are not sure, look on the back panel of the synth. There should be two 5-pin DIN connectors; one marked MIDI IN, and the other MIDI OUT or THRU. They will look just like the matching MIDI ports on the back of the ST (located right next to the power plug). Next you need a MIDI cable. You could probably make one, but they are inexpensive (about $5) since they have few wires in them, so buy or borrow one. Normally you use two, but this demo only plays, so plug one end into the MIDI OUT of the ST and the other end into the MIDI IN on the synth. Plug the audio output from the synth into an amplifier, and you're ready to go.

Turn the power on (synth, amp, and don't forget the ST) and boot MIDIDEMO.PRG. A menu will appear listing the songs. Select one with the mouse or enter the name from the ST keyboard, hit return, then sit back and listen to the music. If you don't hear anything, make sure that all cables are connected properly and that the power is on for all devices. Also the synth and amp have volume controls which may need to be adjusted. Some amps may have a standby switch which should be set to its 'on' position. If you wish to stop the song before its over, hit <esc>. Placing the cursor on the cancel box of the menu and clicking exits from the program.

Easy as that, you've taken your first step into the wide world of MIDI. But of course, you've only scratched the surface (see this month's MIDI Beat). Next month, I'll try to include a program that will allow you to record songs yourself.

I'd like to again thank ATARI for letting us include this progam. It is not in the public domain, but ATARI would like all ST owners to see that MIDI does work, so feel free to pass this demo with all the necessary files to any of your friends with STs or to demo it at your user group.

An Introduction to Telecommunications

by Bill Petry & Bill Marlin

Telephone lines operate with a signal bandwidth of from 300 to about 3000 Hertz (Hz, cycles per second). Direct current (DC) and 20 Hz alternating current (AC) are both used to get the job done. While the telephone is patiently waiting for someone to call, a 0 Hz direct current voltage is constantly being appled to the receiver by the telephone company. In your telephone is a capacitor which is coupled, in series, to the ringer. This capacitor will not allow the direct current to pass. Otherwise the bell would ring constantly. When someone calls you, an alternating current is applied to the wires, which passes through the capacitor and rings the bell. When you pick up the receiver you close the current's path to the ringer and open the path thru the handset for your voice (DC). Your voice tones are then transmitted thru the phone lines to your caller.

Your computer, on the other hand, transmits data by variable voltages. Internally, the computer uses 0 volts for the zero voltage reference against a +5 volts logical '1'. The RS232 standard port converts these 0 to +5 voltages to -12 and +12 volts. However, the phone company really doesn't need for all of us putting these voltages over the wires. Besides that, there isn't a continuous DC path from your phone to another person's phone anyway. A logical solution is to convert these voltages to tones. To do this we use a MOoulator-DEModulator (Modem). The outgoing voltages at the RS232 port on the back of the Atari ST are first converted to audible tones by the modulating circuitry in your modem. These can then be transmitted over the phone lines. At the other end, the demodulating circuitry in the modem there converts these tones back to RS232 compatable voltages which can be utilized by the other computer.

With the Atari 800, the 850 interface is required to further convert these RS232 voltages (in the -12 to +12 range) to the 0 to +5 TTL voltages the 800 was designed to work with.

The actual frequency of the tones varies with the data transmission rate (BAUD). This is measured in bits per second. Currently, the usual baud rate is 300. This is only half as fast as Atari's cassette rate. It is quite slow. The up and comming standard is 1200 baud, with twenty-four hundred (2400) rounding the corner.

From 0 to 300 baud, Frequency-shift keying (FSK) is the method of modulation used. The 1200 baud modems use differential phase-shift keying (DPSK). For the sake of simplicity we'll use the 300 baud type for our example.

At 300 baud, the originating modem (yours), uses 1070 Hz for a logic '0' (SPACE) and 1270 Hz for a logic '1' (MARK). The BBS at the other end answering the call uses 2025 Hz for logic '0' (SPACE), what you hear when the phone is answered by the BBS, and 2225 Hz for the logic '1' (MARK).

These tones, for set periods of time, indicate the value of the data bit that is being transmitted. The example below shows how the letter 'A' is represented:

ASCII	Hexadecimal ($)	Binary
A	41	0 1 0 0 0 0 0 1

The computer will package the byte 'A' thus:

1 0 1 0 0 0 0 0 1 0 ....

The first '1' is the start bit. It tells the receiving computer that a data byte is on the way. At precise time intervals the receiving computer will 'sample' the incoming package. Where it finds a logic '0' it assigns a bit value of 0, a logic '1' it assigns a bit value '1', thereby building up an 8 bit binary number. The trailing '0' (or 0's) is the stop bit(s), which can be very long(i.e. quite a lot of them). This is the same principal of serial transmission used with serial printers.

This is all fine and good when the signal is free of distortion. However, as anyone even slightly familiar with talking on the telephone knows, reception can be quite distorted at times. This quality lead to the concept of 'parity'. No parity exists when you use the full 8 bits of each binary byte you send. Remember, we have one start bit, 8 data bits, and one or more stop bits to each data packet. When we use parity we can set the hi or 7th bit of the data byte to either a set value or one depending on the number of logic 1's present. We can set the 'parity bit' to be always '1' or always '0'. This way we know that at least one bit in the transmission should always be a given value. Again, this is ok for teletypewriters and computers only a few feet apart. However, this only gives us a 1 in 8 check on what we take for reliable data (that's only a 13% verification).

Another way is to equate the parity bit with the number of 1's in the byte. Two possibilities arise: even or odd parity. With even parity we get a 1 if the number of 1's in the byte is even, a 0 if not. The reverse would hold with odd parity, a 1 if the number of 1's is odd, a 0 if the number is even.

This kind of verifying our data would be useless when transmitting more than 128 different characters. Remember the ATASCII character set with the inverse video and control characters? Sending these require the entire 8 bits to be used. One way of getting a checksum with the full 256 available bytes in use (using all 8 bits in each binary number), is what's called X-modem protocol. Here, with every packet of 128 bytes, a checksum is calculated by the receiving computer and this value is then sent back to the originating computer and checked against the checksum calculated on the data before it was sent. If it jives, all's well. If not, the packet is resent and reverified as many times as the software permits until a valid checksum is had. This, in itself, does not guarantee that the data is correct. It just means that the same number for the checksum was received back at the origin.

You can get all the information you need on X-modem protocol by logging on to Ward Christensen's BBS in Chicago, Il. The number to CBBS is (312) 545-8086.

Modems are capable of transmitting data in both directions (remember the different frequencies?), at the same time. In Full-duplex (echo-plex), the modems chat at the same time(like you and your friends on the phone). Your screen shows you only what the other computer echoes back to you. Half-duplex, on the other hand, transmits in one direction at a time, and so is slower. What you see is the echo of what you sent from the receiver's computer. When you are in half-duplex and the other computer is running in full-duplex,everything you see on the screen is doubled, so when you type "HELLO", you see "HHEELL-LLOO". To correct this, all you need to do is switch to full-duplex the next time you call up that bulletin board.

Now that you have an idea of just what's going on with your modem, the next step is to get a modem and hook it up to your ST.

The only thing you need to connect a modem to your computer is a modem cable. A few cautions are in order should you want to try and make the cable yourself.

The first thing to do is go to page 76 in your ST owner's manual, and take a good look at the drawing of the modem port connector and find out which pin does what. Then, find your modem's user's guide and see what each pin on its RS232 port is for. I have an Avatex 1200 and will use it as an example.

I constructed the following chart of the pins for both the computer and modem in order to see which pin needs to connect to which:

Atari ST				Avatex 1200	
Function	Dir.	Pin	Pin	Function	Dir.
Prot. ground	---	1	1	Not connected	---
Trans. data	out	2 --->	2	Trans. data	in
Receive data	in	3 <---	3	Receive data	out
Request to snd	in	4	4	Not connected	---
Clear to send	in	5 <---	5	Clear to send	out
Not connected	---	6	6	Data set ready	out
Signal ground	---	7 <-->	7	Signal ground	out
Data carr. det	in	8 <---	8	Carrier detect	out
			18	Analog loopback	in
Data term. rdy	out	20 --->	20	Data term. ready	in
			21	Remote dig. loop	in
Ring indicator	in	22 <---	22	Ring indicator	out

Looking in a friend's Hayes 1200 Hardware Reference Manual gives an additional entry for a High Speed Indicator at pin #12. Another modem shows +12 volts at #9, -12 volts at pin #10, transmit clock for pin #15 and a receive clock at pin #16. The point I'll make is that you must be sure that you connect only the pins you need for your modem/computer system. With a little negligence you could very easily short circuit a part of your modem, making life just a bit more difficult than it needs to be.

By the way the connector you see at the modem port on your ST, and on your modem are RS232 DB-25 connectors. The computer one is male (with the pins sticking out), and the modem connector is female (for the pins to fit in).

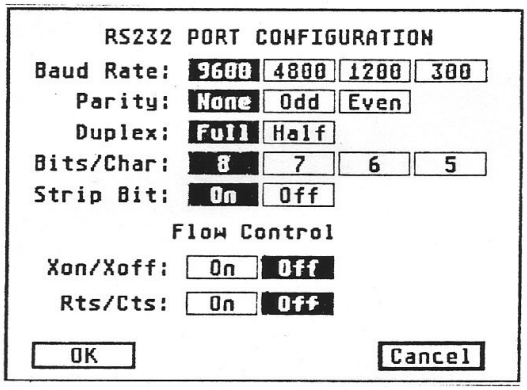
VT52

With your modem and computer connected, your phone bill paid and the number of your local BBS in front of you, you need only boot up TOS and select the VT52 Emulator option at the right of your desk-top.

What this gets you is a terminal emulator which is compatable with a Digital Equipment Company (DEC) VT-52. This is an old reliable form of terminal emulator which was developed as a work station on large DEC computers. It will behave much like the airport reservation terminals where the computer may be downstairs or across the country.

With it you can access files and upload messages. You won't be able to capture files. However, you can save what you see on the screen, to your Epson or IBM PC compatable printer by pressing the 'Alternate' and 'Help' keys at the same time.

Press the HELP key to configure your terminal. Now you'll get the RS232 port configuration window. Anything look familiar?



The only things we see that we haven't mentioned are: Xon/Xoff and Strip bit. Strip bit is the parity bit. Xo/Xoff is mostly used with teleprinters. The print head as it finishes a line of type, must return to the left side of the platten and the paper must advance one line. With Xon the computer sends a request not to send while the above mechanical operation takes place. Otherwise, you'll lose the incoming data.

If you have an acoustic modem, (Atari 830), you need only dial the BBS number, listen for the high-pitched tone, and place the receiver in the cups. You will then see the BBS introductory text feed onto the screen. VT52 has its uses. However, to transfer files to and from ST we must have more sophisticated software. Fret not its already here.

STerminal 1.0

STerminal 1.0, by Jeremy E. San, of Argonaut Software Ltd, London, England is just the ticket to get us our much wanted file transfers in motion.

This software is in the Public Domain and so can't be sold. However a distributor can recover 'media costs, etc.'.

The program is short, just over 2 kilobytes in length, and works 'just fine' capturing text files. You won't be able to download binary files (those with the 8th bit in use) but that's not a serious handicap. Unless a program was specifically written for the ST, it wouldn't run anyway. Source codes are 'text files' and that's what we need right now.

Binary files could be easily converted to their ASCII equivalents and then be shipped as text files to be reconverted back to executable code. However, BASIC is the missing link. And with the RAM memory requirements of TOS and the 'bugged' BASIC that's around, together, we'd have only about 3K of useable RAM for the program and convertable file....definitely not a plus. So we're not a bad off as we might suspect.

STerminal is menu driven and boots with the following lines:

```
STerminal 1.0 by Jeremy E. San
Argonaut Software Ltd, London, England

-PUBLIC DOMAIN- 13th August 1985

Press SHIFT-ESCAPE for Extended Menu
```

Pressing the Shift- and Escape keys at the same time gives the next pair of lines:

```
STerminal Extended Menu.
Press H (help), or: B,C,D,E,I,L,Q,S,U ?
```

Pressing H gets you to the HELP menu:

```
HELP, -Press a single letter as follows:

B = Set the Baud rate,
C = Clear the Capture Buffer,
D = Download test into Capture Buffer,
E = Exit from here back to chat mode,
I = Information about the Buffer,
L = Load a file into Capture Buffer,
Q = Quit STerminal,
S = Save Capture Buffer to a file,
U = Upload text from Capture Buffer.

STerminal Extended Menu.
Press H (help), or: B,C,D,E,I,L,Q,S,U ?
```

Pressing 'B' gives the following range of baud rates to choose from: 19,200, 9,600, 4,800, 3,600, 2,400, 1,800, 1,200, 600, 300, 200, 150, 134, 110, 75, and 50.

Selecting 'E' gets you into the Terminal mode, from which you can return to the extended menu by pressing ESC and SHIFT at the same time.

'L' gets the prompt "Filename?". An error here gets the following response:

You sure it exists??
I'm having trouble locating it!

A gander at the 68000 source code for STerminal (included on the disk version) gives a humorous side to programming. It's reminiscent of that found in the many resources making up the Atari ST Development Package. Despite the mechanical subject matter, those of you who are reading this now are human. That makes it 'Person to Person'.

CHAT

A more sophisticated telecommunications package is already available: 'CHAT', from SST systems, P.O. Box 2315, Titusville, Florida, for $19.95.

CHAT supports autodialing and X-modem transfers. It is, however, not 100% compatable with all BBS's. The reason for this is that some of the old Apple BBS's have a transfer protocol that, although called XMODEM really isn't. CHAT works only with systems that use the real XMODEM protocol as defined by Ward Christensen, its author. With these variant BBS's, you can still capture text files. Also, CHAT uses only pins 2,3, and 7 to talk to the modem. You don't have to be concerned about the others.

Next month we'll give CHAT a run thru its paces, and take a walk thru the local modem shop.

Desktop

Getting Aquainted With Your GEM DESKTOP

Part 1: Setting Up Your printer

The first thing I did with my ST, besides unpacking it & formatting diskettes was assembling a printer cable. I could have purchased one mail-order for around $20, but my adventurous spirit had other plans.

To build it yourself, you need a 36 pin Centronics parallel connector (to fit into the printer slot), a 3' length of 32 strand ribbon cable, and a Male DB-25 conector with hood(its protective covering).

Once you have the pieces together, you need to find the page with the pin configurations so you can tell which pin does what. Then, turn to page 76 of your ST owner's manual. The top illustration shows shows the pinouts with a picture of the female connector that's at the back of your ST.

If you have the press-fit Centronics connector, you need only insert the cable and carefully clamp one end of the ribbon cable between the two end pieces (a small vise with wood inserts helps).

Otherwise you'll need to solder a wire to each connector in use. The connectors usually have the pins numbered, so you only have to match numbers.

You'll need to make only 11 connections, as some of the pins aren't in use. You'll need to connect the strobe (timing pulse), the 8 data lines the busy, and a ground. The pins of the Centronics connector are numbered 1 to 18 on the long side, and 19 to 36 on the short side. This means that the strands in the cable are numbered 1, 19, 2, 20, 3, 21,...coming out of the Centronics connector. Bare the correct wires and solder them to the corresponding pins on the DB-25. Use an ohmmeter to make sure each pin of the DB-25 really goes to the Centronics contact you want it to.

Install the protective hood to the DB-25 and connect your printer to your ST. If all was done correctly you now can print out text files etc.

In order to dump your screen displays your printer must be an EPSON or IBM compatable printer. The Gemini 10 & 10X will print text, but not the graphics. You'll need a Star SG-10, running in PC mode to get that. All EPSON printers work fine excepting the daisywheel varieties.

Desktop continued on page 21

Available Software

This column gives the currently available software for the Atari ST, and where it can be obtained. (11 Sep 1985)

Developers Package for the ST
Atari Inc.
Attn: Richard Frick
1196 Borregas Avenue
Sunnyvale, California 94086
(408) 745-4926
Price $300, +$21 Calif. Sales Tax

This software development package consists of over 1200 pages of documentation and 5 full 3.5" diskettes,

Software:
Utilities, Compiler, Linker, RCS (Resource Construction Set), An upgraded TOS, & MicroEMACS Editor.

Documentation:
Non-Disclosure Agreement (Send back signed copies or you will not get updates)
Introducion to GEM Programming
GEM VDI Programmer's Guide, Vol. 1
GEM AES Programmer's Guide, Vol. 2
GEM DOS Specification, Rev. 13
Example of GEM Program Listing (IBM Version)
DRI "C" Language Programmer's Guide
DRI Programmer's Guide (Assembler, Linker, and Debugger)
ST BIOS Listing
Hitchhiker's Guide to the BIOS
ST Hardware Specifications
KERMIT User's Guide
KERMIT Protocol Manual
Intelligent Keyboard Spcification
ST Schematic
ST to IBM Keyboard Translation Suggestions
Line "A" Manual
Programmable Sound Generator
MicroEMACS Screen Editor Tutorial

Express Letter Processing Package
Mirage Concepts, Inc.
4055 W. Shaw Ave #108
Fresno, California 93711
(800) 641-1442 (To order)
$49.95

Chat, A Telecommunications Package for Atari ST Computers. Ver. 1.0 (Supports X-modem)
SST Systems
P.O. Box 2315
Titusville, Florida 32781
$19.95

Zork I, And The Hitchhiker's Guide to the Galaxy have been seen for sale at Access to Software, 4720 Geary Bl., San Francisco, California (415) 751-2231 (these arrived Monday, 09 Sep 85)

4xFORTH
Dragon Group
148 Poca Fork Road
Elkview, WV 25071
(304) 965-5517
$99.95

ST User Groups

We have been currently informed of only one Atari ST user group.

C.O.A.S.T.
Box 20481
Piedmont Station
Oakland, California 94620-0481
(415) 658-3159

Coming Attractions

SEPTEMBER 1985

Articles and Projects you will find
in upcomming issues of

S/T SIXTEEN/THIRTY-TWO Applications

The Well: Annotated listings of useful programs that are available in other periodicals.

Pascal tutorial series by Chris Robertson.

ST User groups of Planet Earth

FOREM ST: Matt Singer's at it again, this time for the ST. Bulletin board software for the ST.

The Mitsubishi 4853, Double sided, Quad-density 5 1/4" drive and the ST. Want 1.5 Mega-byte storage for under $200?

CHATting with your modem. A commercial telecommunications package for the Atari ST.

A close look at the Atari ST Development Package.

And, of course, the continuations of the 68000, FORTH, and MIDI columns.

AND!

Articles from YOU! the humans for whom this publication is written.

Book Review

Presenting the ATARI ST
by English and Walkowiak
$14.95 from ABACUS Software

Translated from German and published in the U.S. in June, 1985, this book is derived from the authors' experiences with a prototype ST. It is ham-strung by the same problem from which we are still suffering, lack of information and software.

The book starts with a brief introduction to 16-bit processors, with the main focus being the 68000. Included is a listing of the 68000 instruction set with examples of the addressing modes. The second chapter covers some of the ST's other chips. Each of the peripherals is discussed, and there are sections on the chips that deal with the keyboard and things like memory access, interrupts, and sound.

Chapters 3, 4, and 5 describe TOS and GEM, but only from an overview. The knowledge gained from reading the explanation of the 'computer as desktop' metaphor could be had as easily by using an ST and mouse. The book finishes with a short discussion of computer languages (which didn't seem to relate directly to the ST), and a quick look at LOGO. This is basically a synopsis of the manual that comes with LOGO.

This book is a kind of mental Chinese food; an hour after I read it, I was hungry again. It doesn't have enough depth to make it a favorite reference. And if you already have an ST, playing with it should make you feel at home with the GEM desktop; it was designed to be easy to use. Perhaps a prospective ST buyer would benefit most from this book but $15 still seems a bit much.

Reviewed by Richard Kaller
Sebastopol, CA

Software Review

Express Letter Processor
Mirage Concepts, Inc.
4055 West Shaw, #108
Fresno, California 93711
(209) 227-8369
$49.95

The last time I used a typewriter was over 2 years ago. Now, with the release of "Express Letter Processor" I'll be able to type on my ST.

I've heard several people rate this package poorly, just from the remarks they've heard (third or fourth hand).

I agree with them and yet, at the same time I feel that part of the initial response from fellow Atarians is our exposure to such word processors as 'Letter Perfect' and 'Atariwriter' and the host of other fine ones available on our eight-bit little brother. Perhaps most of us haven't been in a working office recently, or have only talked to the secretaries who 'type' most of the day on their typewriters. I have typed in the manuscript of a small book with 'Letter Perfect' and really never got the impression I was typing with a typewriter.

'Express Letter Processor' is the closest thing I've seen to an electric typewriter and still not be an electric typewriter. To grade it on the merits of 'Letter Perfect' would be grossly unfair.

What 'Express' gives you is an electronic piece of paper which can be typed on, edited, saved, and printed. As far as being that it does well. The only software available to compare it to are the MicroEMACS and Mince editors. I have a hard time remembering to hit return at the end of every screen line with MicroEMACS. My text usually runs off the screen. With the Express' wraparound feature I don't have to give it a thought. Once the margins are set my text stays between them comfortably.

There aren't a lot of features, so there aren't a lot of commands to get bogged down with. At the right-hand top of the screen is printed 'HELP KEY'. When you need to remember a feature or command, you need only hit 'Help' and a pair of window-shades roll down over your text.

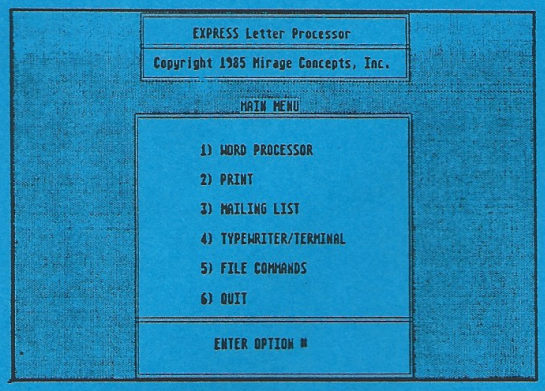
The options available are all control letters with the exception of F1 (function key #1), and Help.

```
PAGE:  1 - LINE:  1                                                HELP KEY
0------A--1---------2---------3---------4---------5---------6---------7-M-----
      January 1, 1990<
      <
      <
      <
      ?A<
      <
      Dear ?F,<
      <
          This is a sample letter which has been included on your<
      "EXPRESS" program disk for the purpose of acquainting you with<
      some of the program's more unique features.<
      <
          It will benefit you to have a printed copy of this letter in<
      front of you as you review this "video version."  To obtain one,<
      exit the =WORD PROCESSING= section, enter the =MAILING LIST=<
      section, enter two or three people (fill =all= fields with<
      information), and print out the letters using "Option M2" from<
      the program's "MAIN MENU".  Re-enter the =WORD PROCESSING=<
      section and refer to the printed version as you read through this<
      letter.<
      <
          First of all, note that in the printed version of this<
      letter a three or four line address was substituted by the<
```

```
PAGE:  1 - LINE:  1                                                HELP KEY
0------A--1---------2---------3---------4---------5---------6---------7-M-----
```

CTRL N - Erase all text	CTRL F - Reform paragraph
CTRL L - Load text from disk	CTRL Y - Remove paragraph
CTRL W - Write text to disk	CTRL J - Insert paragraph
CTRL C - Set margins	CTRL U - Underline print - ON/OFF
CTRL R - Release margin	CTRL B - Bold Face print - ON/OFF
CTRL T - TABS - Set/Remove	CTRL P - Pause here when printing
CTRL V - Delete line	F1 - Quit to main menu
CTRL O - Go to beginning of text	
CTRL K - Go to end of text	HELP - Return to word processor

Delete and Backspace behave quite differently. Backspace does just that. It moves the cursor to the left one space erasing the letter that was there. Delete, on the other hand vacuums text from the right one character at a time.



A feature that really impressed me was 'Reform Paragraph' <CTRL F>. When I was transfering some of the articles in S/T Applications which were written with Atariwriter, I found it impossible for MicroEMACS to cope with the extremely long lines of text. 'Reform Paragraph' broke these long chains into bite-sized pieces, which could then be re-aligned and cleaned up. However, you really should first put a carriage return between the lines before you hit <CTRL F>. Otherwise the entire document is reformatted and that's almost the time to drag the whole thing over to the trash can.

You cannot print your document from the word processor proper. You need to save your file, exit and then enter the 'Print' utility.

When printing you can select a 'Pause' between pages of text if you use single sheets of paper, which gives ample time for alignment etc.

Another feature is 'Merge Print' with which form letters can be printed with names & addresses from the mailing list utility. You simply enter all the data you want in the template provided in the mailing list editor and save your list to the disk. Then, when you write your form letter, you place a '?' followed by the number of the field you want to include and, Wella! there it is.

I haven't tried it yet, but it might even print out mailing labels.

Two other features 'Express' has are the 'Typewriter' and 'Terminal' modes. 'Typewriter' mode is a one-line-at-a-time, semi-intelligent typewriter. After you've composed your line, you hit 'Return' and it prints to the printer. This is nice when printing addresses on envelopes.

The 'Terminal' is somewhat simple in that it only supports capturing of text files. However, this is really all you need when working with a typewriter.

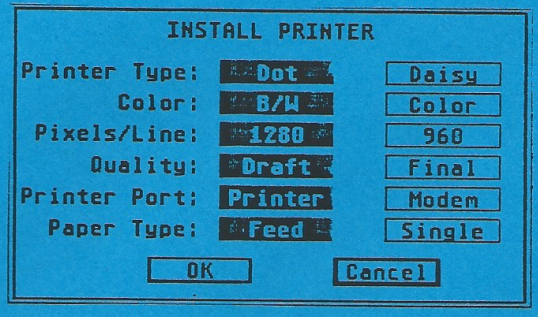
To recommend 'Express' to a hard-core 'Letter Perfect' user, or someone waiting for GEM write would be folly. However if you are used to a typewriter you will find it familiar and with enough features to keep you going for some time to come.

Reviewed by
Bill Petry
13 Sep 1985

More Rumors

Batteries Included, 30 Mural Street, Richmond Hill, Ontario, L4B 1B5 Canada. (416) 881-9941: HOMEPAK is due to be released at the end of this month. DEGAS (Design & Entertainment Graphics Arts System is due to be available in late October. This is a Paint, Draw & Graphics package. ISGUR Portfolio system, an investers management tool is scheduled to be out later this year.

Desktop
continued from page 17



A few tricks you might find useful include using 'Alternate & Help' at the same time to print a screen, and installing your printer with 960 pixels/line to get the entire screen on your sheet of paper.

Take a look at the Install Printer option under the 'Desk' heading at the top of your desktop, and check out the features GEM has available.

Print Quality refers to line spacing. Draft gives double- whereas Final gives single-spacing. Paper type 'Single' gives a pause between sheets, and can be used to pause between pages with tractor feed also.

S/T